Bachelor of Software Engineering - Game Programming

GD2P02 – Physics Programming

Introduction to Physics

Overview

- Introduction to Physics
- Terminology and Core Concepts
- Physics Engine Overview

Where are we today? Why physics?

Introduction to Physics

- Physics is a natural science.
  - Study the rules that govern the natural world…
  - Through scientific methods.
- Physics involves the study of matter and its motion through space and time.
  - Matter: A substance that has **mass** and **volume**.
  - Motion: A **change in position** with respect to time.
  - Space: **3D extent** in which objects and events have position and direction
  - Time: A **dimension** in which events can be **ordered**.

Introduction to Physics continued…

- Some definitions to remember
  - Natural World: The phenomena of the physical world, ranging in scale from sub-atomic to cosmic!
  - Scientific Methods: The techniques for investigating phenomena.
    - Scientific inquiry: No theory can ever be considered completely certain since it always remains subject to falsification.
    - Empirical Evidence: Knowledge acquired by observation or experimentation.
    - Measurable Evidence: measurable results from replicable experiments: quantitative over qualitative
      - » Standardized measurement: SI system
      - » Fundamental units: kilogram, meter, candela, second, ampere, kelvin, mole…

Introduction to Physics continued…

- Some definitions to remember
  - Mass: A coherent, typically large body of matter with no definite shape.
    - Inertial Mass: A quantitative measure of an object's resistance to acceleration.
  - Volume: A quantity of three-dimensional space enclosed by a closed boundary.
  - Position: Spatial location of an entity or object.
  - Direction: Information regarding the relative position of a point with regards to another point.

Introduction to Physics continued…

- Balance, Force and Energy
  - Balance: State of rest
  - Force: An influence that causes an object to undergo a certain change.
  - Energy: The ability of a physics system to do work.

- Scientists study physics in order to understand how the universe behaves…
  - General analysis of nature!
- We, as game programmers, study physics to employ realistic gameplay.

Introduction to Physics continued…

- Physics programming is not easy…
  - Many different ways to solve problems!
    - Lots of “complicated” techniques…
      - For the same problem!
    - There is no one-size-fits-all approach!
  - Some solutions are very heavy on mathematics…
    - Patience!
  - Good to keep in mind: “Start simple”
- The First Law of Video Game Physics:
  - Reality is overrated.
  - Fake reality

Physics in Games

- Why use physics in a game?
  - Dynamic
  - Interactive
  - Responsive
  - Realistic
  - Life-like
  - Flexible
  - …
- Interesting and exciting effects!
  - Games can seem really alive with physics effects!

Physics in Games: Challenges

- Mathematics and Programming
  - Pencil and paper solutions…
  - Long and complicated methods…
  - The more you work at it… the better you will get!
- Computing Power
  - Simulation complexity…
    - Range of hardware: Mobile, Console, PC…
- Applications of fundamental principle
  - We will learn principles as the course progresses

Where is Physics Used in Games?

- Collisions
  - Resting
  - Dynamic
- Bridges
  - Springs
  - Cranks
  - Levers
- Joints
  - Hinges
  - Doors

Fig 1: Bridge Project (Chronic Logic, 2002-2009)

Fig 1: http://www.pcgamer.com/2012/11/25/sim-plicity-i-am-a-bridge-builder/

Where is Physics Used in Games?

- Animation
  - Ragdolls
  - Characters
  - Kinematic behaviour
- Car and Vehicles
  - Wheels
  - Bodies
  - Springs
- Hair
- Cloth
- Water

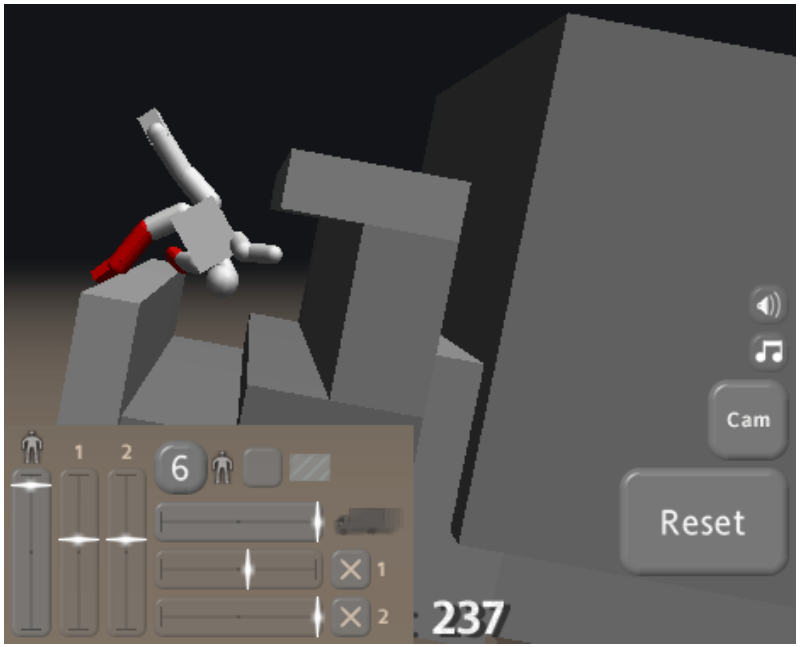

Fig 2: Truck Dismount (Jetro Lauha, 2003)

Fig 2: http://shawnpatton.com/fatstupidduck/dismount.html

What is a Rigid Body?

- Rigid Body
  - Solid object of finite size which cannot deform.
    - The distance between two points on a rigid body will always remain constant regardless of any actions exerted on the body.
  - In the real world, do rigid bodies exist?
    - No… even though some matter seem rigid…
    - But we can assume they do for our simulations.
      - Unless things move near the speed of light…

Simulating Rigid Body Systems

- Penalty Methods
  - A class of algorithm for solving constrained optimisation problems:
    - Constraint: A condition to be satisfied towards a solution of the problem.
    - Optimisation: Selecting the best element from a set of available alternatives…
      - For example: Maximisation or Minimisation of a function!
  - Springs and dampers
    - Lots of springs and forces are combined with dampers and limiting factors to achieve the desired effects and try to improve stability…

Simulating Rigid Body Systems continued…

- Impulses
  - Instantaneous velocity changes
    - Multiple iterations to settle and converge on a solution
    - Additional “tricks” to ensure stacks of objects settle…
    - Objects should use a restitution coefficient…
      - “Bounciness”
    - Bias can be added to penetration problems
    - Shock propagation

Simulating Rigid Body Systems continued…

- Solvers
  - Lagrange Methods
    - Global analytical analysis of a system to determine the exact forces to produce the desired effect
  - Various methods:
    - Featherstone's method
    - Linear algebra methods

Speed vs Accuracy vs Reliability

- Make the technology work…
  - Before optimising!
  - Keep it simple!
    - Now is not the time to try “fancy” programming techniques!
- Make it stable…
  - Then try to speed it up…
- Use sanity checks…
  - Usual good programming techniques!
- Do not assume anything…
  - Use good debugging techniques!

Test Cases

- Create test scenarios to trial physics code…
- If we built a rigid body simulator:
  - Example test cases:
    - See-saws
    - Stacks of blocks
    - Objects on slopes
    - Resting objects
    - Interaction of objects of different shapes
    - Dealing with intersecting objects
    - Walls
    - Chains of objects
    - Etc…

Parts of a Game Physics Engine

- High-level view:
  - Collision Detection:
    - Broad to Narrow
    - Pairs of objects that are intersecting…
  - Clusters:
    - Islands
    - Groups of objects intersecting…
  - Solvers:
    - Work on a cluster to resolve collision response and constraints
  - Integrators:
    - Moving the system forward in time…
      - Discrete time-step.

Parts of a Game Physics Engine

- Core:
  - Soft bodies
  - Rigid bodies
  - Particles
    - Cloth, Hair, Fur…
- Solution:
  - Penalty
    - Springs
  - Impulses
  - Constraint Solvers
    - Iterative, Gaussian Elimination, Matrices…

Parts of a Game Physics Engine

- Movement:
  - Linear Motion
  - Angular Motion
- TimeStep:
  - Fixed (Frequency? 100hz, 30hz?)
  - Variable
- Integrators:
  - Euler
  - Midway
  - RK4
  - Verlet

Parts of a Game Physics Engine

- Goals:
  - Accuracy
  - Stability
  - Speed
- Shapes:
  - Points
  - Convex:
    - Spheres
    - Basic Shapes:
      - » Cubes, Capsules, Hexagons…
  - Concave:
    - Complex Methods…

Parts of a Game Physics Engine

- Enforcement:
  - Collisions:
    - Dynamic collisions
    - Static contacts
  - Constraints:
    - Contacts
    - Joints:
      - Angular
      - Linear
- Complexity:
  - Simple
  - Complex

Parts of a Game Physics Engine

- Problems:
  - Jitter
  - Stability
  - Handle Penetration
  - Stacking
  - Large number of active bodies…
- Lifetime of Objects:
  - Short
  - Long

Parts of a Game Physics Engine

- What about the target platform?
  - PC, Console, Mobile
  - CPU or GPU
  - Memory Usage
  - Architecture
    - Low Level Access

- No one-size-fits-all solution!

High Level Overview

- Pseudo Code:

```
Gameloop
{
    Setup()
    while (!gameover)
    {
        simPrePhysics();
        physicsEngine->Update();
        simPostPhysics();
        Render();
    }
}
```

Preparation

- Need a basic renderer to…
  - DrawPoints(Vertices, Count, Colour)
  - DrawPolygon(Vertices, Count, Colour)
  - DrawCircle(Center, Radius, Color)
  - DrawDebugText(Text, Color)
  - SetOrtho(…)
    - Parallel Projection…

Summary

- Introduction to Physics
- Terminology and Core Concepts
- Physics Engine Overview